

5. (Base) Graphics

Common parameters

- `formula`, `data`: a formula of the form `y ~ model` and a data frame containing the variables
- `main`, `sub`; `xlab`, `ylab`: main title, subtitle; *x*-axis, *y*-axis labels
- `xlim`, `ylim`, each a 2-vector (low, high): *x*-axis, *y*-axis limits
- `pch`: plotting character (see `?points`)
- `cex` (symbols), `cex.axis`, `cex.lab`, `cex.main`, `cex.sub`: character expansion (relative to 1)
- see `?par` for others

Numeric data

- `boxplot(x)` makes a boxplot from vector `x`; `boxplot(x ~ g)` groups by factor `g`; e.g.

```
boxplot(mtcars$mpg, main="Gas mileage", ylab="miles per gallon", ylim=c(0,40))  
boxplot(mpg ~ factor(cyl), data=mtcars, xlab="cylinders", ylab="miles per gallon")
```
- `stripchart(x, method="overplot")` makes a dot plot of `x` (better than boxplot for small sample); `stripchart(x ~ g)` groups by `g`; `method` handles duplicates: "overplot", "jitter", or "stack"; e.g. `stripchart(mpg ~ factor(am), data=mtcars, method="stack")`
- `hist(x, breaks="Sturges", freq=NULL)`, makes a histogram from `x`, where `breaks` is a vector of bin boundaries (or, as in the default "Sturges", the name of a bin algorithm); `freq=FALSE` gives density histogram instead of frequency; e.g. `hist(mtcars$mpg)`
- `plot(x, y)` makes a scatterplot from vectors `x` and `y`; e.g. `x = 1:5; y = 2*x; plot(x, y), plot(x, y, xlim=c(0,10), ylim=c(0,10))`
- `points(x, y)` adds points to a plot, and `lines(x, y)` adds line segments; e.g. `points(x, x, pch=15); lines(x=c(1,3,5,7,9), y=c(8,1,4,1,8), col="red")`
- `plot(density(x))` makes a *density plot* (usually better than a histogram) from `x`; `rug(x)` adds the data points; e.g. `plot(density(mtcars$mpg)); rug(mtcars$mpg)`
(note: `density(x)` estimates density $f(x)$, returning a list including `(x, y)`, where $y \approx f(x)$)
- `pairs(x)` makes a matrix of scatterplots of pairs of columns of data frame `x`; e.g. `pairs(mtcars)`
- `curve(expr, from=NULL, to=NULL, n=101, add=FALSE, type="l")` draws a curve of `expr` over `[from, to]` (`add=TRUE` \implies add to existing plot); e.g.

```
curve(expr=x*sin(1/x), from=-pi/6, to=pi/6, n=200); curve(expr=x*1, add=TRUE, col="red")
```

Legends; math expressions in titles and labels

`legend(x, y, legend, col=par("col"), lty, pch)` makes a legend at (x, y) (or x can be one of {"bottomright", etc.}: see `?legend`) using labels, colors, line types, and plotting characters in vectors `legend`, `col`, `lty`, and `pch`; e.g.

```
legend("top", legend=c("x*sin(1/x)", "x"), col=c("black", "red"), lty=c(1, 1))
```

Use `expression(...)` in character string used as main or `xlab` or `ylab`; see `?plotmath`. e.g.

```
legend("top", legend=c(expression(x*sin(frac(1,x))), "x"), col=c("black", "red"), lty=c(1, 1))
```

Categorical data

- `barplot(height, names.arg = NULL)` makes a barplot of the counts in `height`, with (optional) bar labels in `names.arg`; e.g.

```
counts = table(mtcars$cyl); barplot(counts)
```

- `mosaicplot(x)` makes a mosaic plot from a table of counts from `table()`; e.g.

```
counts = table(mtcars$cyl, mtcars$gear); mosaicplot(counts)
```

Multiple figures

`matrix(data, nrow, ncol, byrow=FALSE)` fills an `nrow` \times `ncol` matrix by column from `data`
`layout(mat)`, for matrix `mat`, divides graph so *i*th figure is drawn where `mat==i` (0 \implies blank)
`layout.show(n=1)` shows outlines of next *n* figures; e.g.

```
m = matrix(data=c(1, 0, 2, 3, 3, 3), nrow=2, ncol=3, byrow=TRUE)
layout(m)
layout.show(3)
hist(mtcars$mpg)           # 1st plot: (frequency) histogram alone
plot(density(mtcars$mpg))  # 2nd plot: density plot alone
hist(mtcars$mpg, freq=FALSE) # 3rd plot: density histogram
lines(density(mtcars$mpg))  # add density plot to (3rd plot) histogram
layout(matrix(data=1, nrow=1, ncol=1)) # reset graphics device
```

Write graphical output to a file

- Open a graphical output file with, e.g., `pdf("file.pdf")`, `png("file.png")`, `jpeg("file.jpg")`, `bmp("file.bmp")`, `postscript("file.ps")`, `tiff("file.tif")`
- Make graph
- Close the file with `dev.off()`